MagicMirror – manuelles Backup auf NAS

Zweck

Der komplette MagicMirror-Ordner soll manuell auf einem NAS gesichert werden.

Das Backup wird nur gestartet, wenn das NAS tatsächlich erreichbar und als CIFS/SMB-Freigabe eingebunden ist.

Das ist besonders wichtig, da das lokale Verzeichnis

```
/mnt/nasbackup
```

auch dann existieren kann, wenn das NAS nicht verbunden ist.

Ohne zusätzliche Prüfung könnte ein Backup-Script sonst versehentlich auf die lokale SSD des Raspberry Pi schreiben.

Das Backup wird:

- ausschließlich manuell gestartet
- im Hintergrund ausgeführt
- mit Datum und Uhrzeit gespeichert
- als vollständiges TAR.GZ-Archiv erstellt
- nach der Erstellung automatisch geprüft
- bei einem Fehler als `_INCOMPLETE` gekennzeichnet
- nicht gestartet, wenn das NAS nicht erreichbar ist
- gegen mehrere gleichzeitig laufende Backups geschützt

Es gibt bewusst:

- keinen automatischen Zeitplan
- keine automatische Löschung alter Backups
- keine automatische Rotation

Warum TAR.GZ und nicht rsync?

Ursprünglich wurde versucht, den kompletten MagicMirror-Ordner mit `rsync -a` direkt auf die SMB-Freigabe zu kopieren.

Dabei traten Fehler dieser Art auf:

```
symlink ".../node_modules/.bin/..." -> "..." failed: Operation not supported
(95)
```

MagicMirror und insbesondere `node_modules` enthalten viele symbolische Links.

Die verwendete SMB/CIFS-Freigabe unterstützt diese Linux-Symlinks nicht direkt.

Deshalb wird der komplette MagicMirror-Ordner jetzt zuerst als TAR.GZ-Archiv geschrieben.

Innerhalb eines TAR-Archivs bleiben:

- Symlinks
- versteckte Dateien
- .git-Verzeichnisse
- node_modules
- Dateistruktur

erhalten.

Das NAS sieht dabei nur eine normale Datei:

```
MagicMirror.tar.gz
```

Schnellstart – vorhandenes System

Backup starten

Das Backup wird mit folgendem Befehl manuell gestartet:

```
~/bin/backup_magicmirror.sh
```

Beispielausgabe:

```
Prüfe NAS ...
NAS ist erreichbar.

Backup wird im Hintergrund gestartet.
Quelle: /home/<PI-USER>/MagicMirror
Ziel:   /mnt/nasbackup/Manuelle_Backups/MagicMirror_2026-10-07_11-28-53
Log:    /home/<PI-USER>/magicmirror-backup-
logs/MagicMirror_2026-10-07_11-28-53.log

Backup-Prozess gestartet.
PID: 12345
```

Das Terminal kann danach geschlossen werden.

Das Backup läuft über nohup im Hintergrund weiter.

Aktuellen Backup-Status ansehen

Das neueste Log anzeigen:

```
tail -f "$(ls -t ~/magicmirror-backup-logs/*.log | head -1)"
```

Mit

```
Ctrl+C
```

wird nur die Log-Anzeige beendet.

Das Backup selbst läuft weiter.

Prüfen, ob ein Backup noch läuft

```
pgrep -af 'backup_magicmirror|tar|gzip'
```

Wenn keine Ausgabe erscheint, läuft aktuell kein Backup.

Erfolgreiches Backup

Bei erfolgreichem Abschluss steht im Log:

```
================================================
BACKUP ERFOLGREICH
Ende: ...

Archiv:
/mnt/nasbackup/Manuelle_Backups/MagicMirror_YYYY-MM-DD_HH-MM-
SS/MagicMirror.tar.gz

Größe:
...
================================================
```

Auf dem NAS befindet sich anschließend beispielsweise:

```
Backup_magicmirror/
└── Manuelle_Backups/
    ├── MagicMirror_2026-10-07_11-28-53/
    │   └── MagicMirror.tar.gz
    ├── MagicMirror_2026-11-15_17-22-10/
    │   └── MagicMirror.tar.gz
    └── ...
```

Fehlgeschlagenes Backup

Während das Backup läuft, trägt der Zielordner absichtlich den Zusatz:

```
_INCOMPLETE
```

Beispiel:

```
MagicMirror_2026-10-07_11-28-53_INCOMPLETE
```

Erst wenn:

- TAR erfolgreich beendet wurde
- gzip das Archiv erfolgreich geprüft hat
- TAR das Archiv vollständig lesen konnte

wird der Ordner umbenannt und `_INCOMPLETE` entfernt.

Bleibt `_INCOMPLETE` stehen, ist dieses Backup nicht als gültige Sicherung anzusehen.

NAS-Mount überprüfen

Ist das NAS wirklich gemountet?

```
findmnt /mnt/nasbackup
```

Bei korrektem systemd-Automount können zwei Zeilen erscheinen:

```
/mnt/nasbackup systemd-1                        autofs ...
/mnt/nasbackup //<NAS-IP>/<SHARE>               cifs   ...
```

Das ist korrekt.

Die erste Zeile ist der systemd-Automount.

Die zweite Zeile ist die tatsächlich gemountete SMB/CIFS-Freigabe.

Speicherplatz des NAS anzeigen

```
df -h /mnt/nasbackup
```

Schreibzugriff testen

```
touch /mnt/nasbackup/test_magicmirror.txt
ls -l /mnt/nasbackup/test_magicmirror.txt
rm /mnt/nasbackup/test_magicmirror.txt
```

Wenn alle drei Befehle ohne Fehler funktionieren, besteht Schreibzugriff auf das NAS.

Einrichtung auf einem neuen Raspberry Pi

Diese Anleitung beschreibt die komplette Einrichtung auf einem frischen Raspberry Pi.

Als Platzhalter werden verwendet:

```
<PI-USER>       Linux-Benutzer auf dem Raspberry Pi
<NAS-IP>        IP-Adresse oder DNS-Name des NAS
<SHARE>         Name der SMB-Freigabe
<NAS-USER>      Benutzer auf dem NAS
<NAS-PASSWORT>  Passwort des NAS-Benutzers
<UID>           Benutzer-ID des Pi-Benutzers
<GID>           Gruppen-ID der Gruppe nasusers
```

Beispiel für eine Freigabe:

```
//<NAS-IP>/<SHARE>
```

1. Benutzer auf dem Raspberry Pi anlegen

Bei aktuellen Raspberry-Pi-OS-Versionen wird normalerweise bereits während der Installation ein Benutzer angelegt.

Falls ein zusätzlicher Benutzer benötigt wird:

```
sudo adduser <PI-USER>
```

Falls der Benutzer Administratorrechte erhalten soll:

```
sudo usermod -aG sudo <PI-USER>
```

Danach kontrollieren:

```
id <PI-USER>
```

2. Gruppe für NAS-Zugriff anlegen

Für den NAS-Zugriff wird eine eigene Gruppe verwendet:

```
sudo groupadd nasusers
```

Falls die Gruppe bereits existiert, erscheint eine entsprechende Meldung. Das ist unkritisch.

Benutzer zur Gruppe hinzufügen:

```
sudo usermod -aG nasusers <PI-USER>
```

Danach neu anmelden oder den Raspberry Pi neu starten, damit die neue Gruppenmitgliedschaft aktiv wird.

Prüfen:

```
getent group nasusers
```

und:

```
id <PI-USER>
```

In der Ausgabe von `id` muss `nasusers` erscheinen.

3. UID und GID ermitteln

Die numerische Benutzer-ID ermitteln:

```
id -u <PI-USER>
```

Die Gruppen-ID von `nasusers` ermitteln:

```
getent group nasusers
```

Beispiel:

```
nasusers:x:1003:<PI-USER>
```

Dann wäre:

```
GID = 1003
```

Die tatsächlichen Werte auf dem jeweiligen Pi verwenden.

4. CIFS-Unterstützung installieren

Für den Zugriff auf eine Windows-/Synology-/SMB-Freigabe wird `cifs-utils` benötigt:

```
sudo apt update
sudo apt install cifs-utils
```

Ein kompletter Samba-Server wird nicht benötigt.

Der Raspberry Pi ist hier nur SMB-Client.

5. Lokalen Mountpunkt anlegen

```
sudo mkdir -p /mnt/nasbackup
```

Dieser Ordner ist nur der lokale Einhängepunkt.

Die eigentlichen Daten befinden sich später auf dem NAS.

6. Zugangsdaten des NAS sicher speichern

Verzeichnis anlegen:

```
sudo mkdir -p /etc/samba
```

Credentials-Datei erstellen:

```
sudo nano /etc/samba/credentials-nasbackup
```

Inhalt:

```
username=<NAS-USER>
password=<NAS-PASSWORT>
```

Optional kann bei Bedarf zusätzlich eine Domain bzw. Workgroup angegeben werden:

```
domain=WORKGROUP
```

Die Datei darf nicht für normale Benutzer lesbar sein:

```
sudo chown root:root /etc/samba/credentials-nasbackup
sudo chmod 600 /etc/samba/credentials-nasbackup
```

Kontrolle:

```
ls -l /etc/samba/credentials-nasbackup
```

Erwartet wird ungefähr:

```
-rw------- 1 root root ...
```

Das Passwort niemals direkt in `/etc/fstab` eintragen.

7. NAS in /etc/fstab eintragen

Vorher Sicherung erstellen:

```
sudo cp /etc/fstab /etc/fstab.backup
```

Datei bearbeiten:

```
sudo nano /etc/fstab
```

Folgende Zeile ergänzen:

```
//<NAS-IP>/<SHARE> /mnt/nasbackup cifs credentials=/etc/samba/credentials-nasbackup,vers=3.0,iocharset=utf8,uid=<UID>,gid=<GID>,file_mode=0660,dir_mode=0770,_netdev,nofail,x-systemd.automount,x-systemd.mount-timeout=10s 0 0
```

Dabei müssen `<UID>` und `<GID>` durch die zuvor ermittelten numerischen Werte ersetzt werden.

Bedeutung der wichtigsten Mount-Optionen

Option	Bedeutung
credentials=...	Zugangsdaten liegen in einer separaten geschützten Datei
vers=3.0	SMB 3 wird verwendet
uid=<UID>	Dateien erscheinen lokal als Eigentum des Pi-Benutzers
gid=<GID>	Dateien gehören lokal zur Gruppe `nasusers`
file_mode=0660	Lesen/Schreiben für Benutzer und Gruppe
dir_mode=0770	Vollzugriff auf Verzeichnisse für Benutzer und Gruppe
_netdev	Kennzeichnet den Mount als Netzwerk-Dateisystem
nofail	Der Pi darf auch starten, wenn das NAS nicht erreichbar ist
x-systemd.automount	NAS wird erst beim ersten Zugriff verbunden
x-systemd.mount-timeout=10s	Mountversuch läuft nicht unbegrenzt

8. systemd neu einlesen

```
sudo systemctl daemon-reload
```

Automount starten:

```
sudo systemctl start mnt-nasbackup.automount
```

Status kontrollieren:

```
systemctl status mnt-nasbackup.automount --no-pager -l
```

Die korrekte Anzeige lautet:

```
Active: active (waiting)
```

`waiting` ist bei einem Automount korrekt.

Das NAS ist zu diesem Zeitpunkt möglicherweise noch gar nicht physisch gemountet.

systemd wartet auf den ersten Zugriff.

9. NAS-Mount auslösen

Ein einfacher Zugriff reicht:

```
ls -la /mnt/nasbackup
```

Dadurch wird der CIFS-Mount automatisch aktiviert.

10. Mount kontrollieren

```
findmnt /mnt/nasbackup
```

Es sollte unter anderem eine CIFS-Zeile erscheinen:

```
//<NAS-IP>/<SHARE> cifs ...
```

Speicherplatz prüfen:

```
df -h /mnt/nasbackup
```

11. Schreibzugriff testen

```
touch /mnt/nasbackup/test_magicmirror.txt
ls -l /mnt/nasbackup/test_magicmirror.txt
rm /mnt/nasbackup/test_magicmirror.txt
```

Erst wenn dieser Test funktioniert, sollte das Backup-Script eingerichtet werden.

Backup-Script installieren

1. Script-Verzeichnis anlegen

Als normaler Pi-Benutzer:

```
mkdir -p ~/bin
```

2. Backup-Script erstellen

```
nano ~/bin/backup_magicmirror.sh
```

Folgenden Inhalt einfügen:

```
#!/usr/bin/env bash

SOURCE="$HOME/MagicMirror"
NAS="/mnt/nasbackup"
BACKUP_BASE="$NAS/Manuelle_Backups"
LOGDIR="$HOME/magicmirror-backup-logs"

# WICHTIG:
# Hier exakt den erwarteten NAS-Share eintragen.
# Dadurch verhindert das Script, dass versehentlich auf
# das lokale Verzeichnis /mnt/nasbackup geschrieben wird.
EXPECTED_SOURCE="//<NAS-IP>/<SHARE>"


check_nas() {

    # Der Zugriff auf den Mountpunkt löst den systemd-Automount aus.
    # timeout verhindert, dass das Script bei einem nicht erreichbaren
    # NAS unbegrenzt wartet.
    if ! timeout 15s stat "$NAS" >/dev/null 2>&1; then
        echo "FEHLER: NAS ist nicht erreichbar."
        return 1
    fi

    # Prüfen, welches Dateisystem tatsächlich hinter /mnt/nasbackup liegt.
    #
    # Ein vorhandenes autofs-Verzeichnis reicht NICHT aus.
    # Es muss wirklich ein CIFS-Mount des richtigen NAS-Shares vorhanden
sein.
    MOUNT_INFO="$(findmnt -T "$NAS" -n -o FSTYPE,SOURCE | tail -n 1)"
```

```
    FSTYPE="$(awk '{print $1}' <<< "$MOUNT_INFO")"
    MOUNT_SOURCE="$(awk '{print $2}' <<< "$MOUNT_INFO")"

    if [[ "$FSTYPE" != "cifs" || "$MOUNT_SOURCE" != "$EXPECTED_SOURCE" ]];
then
        echo "FEHLER: NAS-Share ist nicht korrekt eingebunden."
        echo "Gefunden: $MOUNT_INFO"
        return 1
    fi

    return 0
}


# ============================================================
# Hintergrundprozess
# ============================================================

if [[ "${1:-}" == "--worker" ]]; then

    FINAL_DEST="$2"

    # Während der Sicherung bekommt der Ordner bewusst
    # den Zusatz _INCOMPLETE.
    TEMP_DEST="${FINAL_DEST}_INCOMPLETE"

    ARCHIVE="$TEMP_DEST/MagicMirror.tar.gz"


    # ------------------------------------------------------
    # Verhindert mehrere gleichzeitige Backups
    # ------------------------------------------------------

    exec 9>/tmp/magicmirror-backup.lock

    if ! flock -n 9; then
        echo "FEHLER: Es läuft bereits ein MagicMirror-Backup."
        exit 1
    fi


    echo
    echo "=================================================="
    echo "MagicMirror Backup"
    echo "Start:  $(date)"
    echo "Quelle: $SOURCE"
    echo "Ziel:   $FINAL_DEST"
    echo "=================================================="
    echo
```

```
# ------------------------------------------------------
# NAS im Hintergrundprozess nochmals prüfen
# ------------------------------------------------------

if ! check_nas; then
    echo
    echo "BACKUP ABGEBROCHEN: NAS nicht verfügbar."
    exit 2
fi



# ------------------------------------------------------
# Temporären Backup-Ordner anlegen
# ------------------------------------------------------

mkdir -p "$TEMP_DEST" || {
    echo "FEHLER: Backup-Verzeichnis konnte nicht erstellt werden."
    exit 3
}


echo "Erstelle vollständiges TAR.GZ-Archiv ..."
echo



# ------------------------------------------------------
# Gesamten MagicMirror-Ordner sichern
#
# Gesichert werden unter anderem:
#
# - config
# - css
# - modules
# - node_modules
# - versteckte Dateien
# - .git
# - symbolische Links
#
# Das Archiv enthält einen obersten Ordner "MagicMirror".
# ------------------------------------------------------

tar \
    --create \
    --gzip \
    --file="$ARCHIVE" \
    --directory="$HOME" \
    MagicMirror

RESULT=$?
```

```
# ------------------------------------------------------
# tar-Fehler behandeln
# ------------------------------------------------------

if [[ $RESULT -ne 0 ]]; then
    echo
    echo "======================================================"
    echo "BACKUP FEHLGESCHLAGEN"
    echo "tar Fehlercode: $RESULT"
    echo
    echo "Unvollständige Sicherung:"
    echo "$TEMP_DEST"
    echo "======================================================"
    exit "$RESULT"
fi


echo
echo "Archiv wurde erstellt."
echo "Prüfe Archiv ..."


# ------------------------------------------------------
# gzip-Struktur prüfen
# ------------------------------------------------------

if ! gzip -t "$ARCHIVE"; then
    echo
    echo "FEHLER: Komprimiertes Archiv ist beschädigt."
    exit 4
fi


# ------------------------------------------------------
# Zusätzlich prüfen, ob TAR das gesamte Archiv lesen kann
# ------------------------------------------------------

if ! tar -tzf "$ARCHIVE" >/dev/null; then
    echo
    echo "FEHLER: TAR-Archiv konnte nicht vollständig gelesen werden."
    exit 5
fi


# ------------------------------------------------------
# Erst jetzt gilt das Backup als erfolgreich.
#
# Der Zusatz _INCOMPLETE wird entfernt.
# ------------------------------------------------------
```

```
    mv "$TEMP_DEST" "$FINAL_DEST"


    echo
    echo "======================================================"
    echo "BACKUP ERFOLGREICH"
    echo "Ende: $(date)"
    echo
    echo "Archiv:"
    echo "$FINAL_DEST/MagicMirror.tar.gz"
    echo
    echo "Größe:"
    du -h "$FINAL_DEST/MagicMirror.tar.gz"
    echo "======================================================"

    exit 0
fi


# ============================================================
# Manueller Aufruf
# ============================================================


# ------------------------------------------------------------
# Prüfen, ob MagicMirror überhaupt existiert
# ------------------------------------------------------------

if [[ ! -d "$SOURCE" ]]; then
    echo
    echo "FEHLER: MagicMirror-Ordner wurde nicht gefunden:"
    echo "$SOURCE"
    exit 1
fi


echo "Prüfe NAS ..."


# ------------------------------------------------------------
# NAS prüfen
# ------------------------------------------------------------

if ! check_nas; then

    echo
    echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
    echo "BACKUP NICHT GESTARTET"
    echo
    echo "Das NAS ist offline oder der Share nicht gemountet."
```

```
    echo
    echo "Es wurden keine Daten lokal nach"
    echo "/mnt/nasbackup geschrieben."
    echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
    echo

    exit 1
fi


# ------------------------------------------------------------
# Ziel- und Log-Verzeichnisse erstellen
# ------------------------------------------------------------

mkdir -p "$BACKUP_BASE"
mkdir -p "$LOGDIR"


# ------------------------------------------------------------
# Datum und Uhrzeit erzeugen
# ------------------------------------------------------------

STAMP="$(date '+%Y-%m-%d_%H-%M-%S')"

DEST="$BACKUP_BASE/MagicMirror_$STAMP"
LOG="$LOGDIR/MagicMirror_$STAMP.log"


echo "NAS ist erreichbar."
echo
echo "Backup wird im Hintergrund gestartet."
echo "Quelle: $SOURCE"
echo "Ziel:   $DEST"
echo "Log:    $LOG"
echo


# ------------------------------------------------------------
# Hintergrundprozess starten
#
# nohup sorgt dafür, dass das Backup weiterläuft,
# wenn das SSH-/Terminal-Fenster geschlossen wird.
# ------------------------------------------------------------

nohup "$0" --worker "$DEST" >"$LOG" 2>&1 &

PID=$!


echo "Backup-Prozess gestartet."
echo "PID: $PID"
```

```
echo
echo "Status ansehen mit:"
echo "tail -f \"$LOG\""
echo

exit 0
```

3. Script ausführbar machen

```
chmod +x ~/bin/backup_magicmirror.sh
```

4. Ersten Test durchführen

```
~/bin/backup_magicmirror.sh
```

Danach Log ansehen:

```
tail -f "$(ls -t ~/magicmirror-backup-logs/*.log | head -1)"
```

Verhalten bei ausgeschaltetem NAS

Wenn das NAS ausgeschaltet oder die Freigabe nicht erreichbar ist, darf das Script kein Backup starten.

Beispiel:

```
Prüfe NAS ...

FEHLER: NAS ist nicht erreichbar.

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
BACKUP NICHT GESTARTET

Das NAS ist offline oder der Share nicht gemountet.

Es wurden keine Daten lokal nach
/mnt/nasbackup geschrieben.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

Das ist ein wichtiges Sicherheitsmerkmal.

Das Script prüft nicht nur, ob `/mnt/nasbackup` existiert, sondern kontrolliert:

- dass tatsächlich ein CIFS-Dateisystem gemountet ist

- dass genau der erwartete NAS-Share gemountet wurde

Wiederherstellung eines Backups

Archiv zunächst prüfen

Vor einer Wiederherstellung kann das Archiv getestet werden:

```
gzip -t /mnt/nasbackup/Manuelle_Backups/MagicMirror_YYYY-MM-DD_HH-MM-SS/MagicMirror.tar.gz
```

Wenn keine Ausgabe erscheint, ist die gzip-Struktur in Ordnung.

TAR-Inhalt testen:

```
tar -tzf /mnt/nasbackup/Manuelle_Backups/MagicMirror_YYYY-MM-DD_HH-MM-SS/MagicMirror.tar.gz >/dev/null
```

Auch hier bedeutet keine Fehlermeldung einen erfolgreichen Test.

Inhalt ansehen

```
tar -tzf /mnt/nasbackup/Manuelle_Backups/MagicMirror_YYYY-MM-DD_HH-MM-SS/MagicMirror.tar.gz | head -50
```

Backup in das Home-Verzeichnis wiederherstellen

Als normaler Pi-Benutzer:

```
cd ~
```

Falls bereits ein MagicMirror-Ordner vorhanden ist, diesen vorher umbenennen:

```
mv ~/MagicMirror ~/MagicMirror_alt
```

Danach Backup entpacken:

```
tar -xzf /mnt/nasbackup/Manuelle_Backups/MagicMirror_YYYY-MM-DD_HH-MM-SS/MagicMirror.tar.gz -C ~
```

Danach sollte wieder vorhanden sein:

```
/home/<PI-USER>/MagicMirror
```

Besonderheit beim Umzug auf einen neuen Pi / neues Betriebssystem

Das Backup enthält bewusst auch:

```
MagicMirror/node_modules
```

Für eine Wiederherstellung auf demselben oder einem sehr ähnlichen System kann das praktisch sein.

Bei einem Wechsel auf:

- einen Raspberry Pi 5
- eine deutlich neuere Raspberry-Pi-OS-Version
- eine andere Node.js-Version
- eine andere CPU-/Systemumgebung

sollten vorhandene `node_modules` jedoch nicht blind als dauerhaft kompatibel betrachtet werden.

Einzelne Node-Module können native Binärbestandteile enthalten.

Für eine saubere Migration auf einen neuen Pi sollte daher zunächst der komplette MagicMirror-Ordner wiederhergestellt werden, damit alle Konfigurationen, Module und Git-Daten vorhanden sind.

Anschließend sollte die Node.js-/MagicMirror-Umgebung passend zum neuen System aktualisiert bzw. die Abhängigkeiten neu installiert werden.

Das Backup dient damit sowohl als vollständige Sicherung als auch als Grundlage für eine spätere Migration.

Nützliche Kontrollbefehle

Automount

```
systemctl status mnt-nasbackup.automount --no-pager -l
```

Tatsächlichen NAS-Mount anzeigen

```
findmnt /mnt/nasbackup
```

Freien NAS-Speicher anzeigen

```
df -h /mnt/nasbackup
```

Laufendes Backup suchen

```
pgrep -af 'backup_magicmirror|tar|gzip'
```

Neuestes Backup-Log anzeigen

```
tail -f "$(ls -t ~/magicmirror-backup-logs/*.log | head -1)"
```

Alle Backup-Logs anzeigen

```
ls -lah ~/magicmirror-backup-logs/
```

Alle manuellen Backups auf dem NAS anzeigen

```
ls -lah /mnt/nasbackup/Manuelle_Backups/
```

Wichtige Dateien

Datei / Pfad	Funktion
~/bin/backup_magicmirror.sh	Manuelles Backup-Script
~/MagicMirror	Zu sichernder MagicMirror-Ordner
~/magicmirror-backup-logs/	Lokale Logs der Backups
/mnt/nasbackup	Mountpunkt des NAS
/mnt/nasbackup/Manuelle_Backups	Ziel der manuellen Backups
/etc/samba/credentials-nasbackup	Zugangsdaten zum NAS
/etc/fstab	Konfiguration des NAS-Automounts
/etc/fstab.backup	Sicherung der fstab

Kurzfassung für den Alltag

Normalerweise wird nur dieser eine Befehl benötigt:

```
~/bin/backup_magicmirror.sh
```

Status ansehen:

```
tail -f "$(ls -t ~/magicmirror-backup-logs/*.log | head -1)"
```

Ein Backup ist erst dann gültig, wenn im Log

```
BACKUP ERFOLGREICH
```

steht und der Backup-Ordner keinen Zusatz

```
_INCOMPLETE
```

mehr besitzt.

From:
https://wiki.mahlen.biz/ - **Smart-Home Wiki**

Permanent link:
https://wiki.mahlen.biz/doku.php?id=magicmirror:magicmirror_tipps:test&rev=1791366282

Last update: **07.10.2026**